



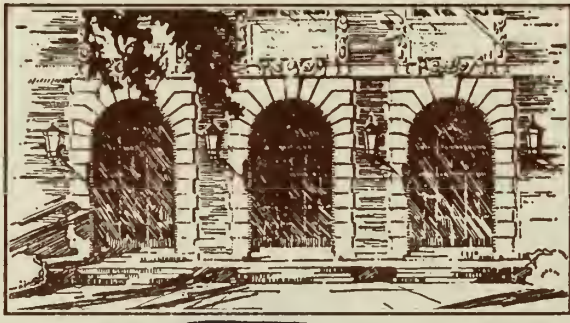
LIBRARY OF THE  
UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN

510.84

Il6r

no. 337-342

cop. 2





Digitized by the Internet Archive  
in 2013

<http://archive.org/details/analogtodigitalc337chel>







Report No. 337

AN ANALOG-TO-DIGITAL CONVERSION  
SUPERVISOR FOR THE IBM 1800

by

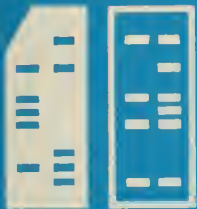
Hartley A. Chelin

June, 1969

THE LIBRARY OF THE

NOV 9 1972

UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN



DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS





Report No. 337

AN ANALOG-TO-DIGITAL CONVERSION  
SUPERVISOR FOR THE IBM 1800\*

by

Hartley A. Chelin

June, 1969

Department of Computer Science  
University of Illinois  
Urbana, Illinois 61801

\* This work was supported in part by the National Science Foundation under Grant No. NSF-GP-7634 and was submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, August, 1969.



5/0.84  
Ilbr  
no. 337-342  
copy 2

iii

#### ACKNOWLEDGMENT

I would like to express my appreciation to Professor H. G. Friedman, who suggested the thesis topic and supervised its development. Also, I am extremely grateful to Mr. Richard A. Wells for his invaluable assistance and to Miss Sandra Claypool, who typed the manuscript.



## TABLE OF CONTENTS

	Page
1. INTRODUCTION . . . . .	1
2. SYSTEM DESCRIPTION . . . . .	3
3. DESIGN CRITERIA . . . . .	6
4. PROGRAM LOGIC . . . . .	9
4.1 Command Phase . . . . .	10
4.2 Execute Phase . . . . .	20
BIBLIOGRAPHY . . . . .	26



## 1. INTRODUCTION

Prior to August, 1968, the ILLIAC II computer provided the University of Illinois with a basic analog processing capability. To extend this capability into third generation equipment, compatible with the University of Illinois IBM System/360 complex, an IBM 1800 computer was acquired. With the implementation of Department of Computer Science written software<sup>1</sup> (i.e., AMOS, the Attached Machine Operating System, on the 1800, and a supervisory program resident on the 360), the 1800 was soon an integral part of the Department's computing system.

At this point, it was possible to do analog processing on the 1800 provided that the user was familiar with assembly language programming, the details of AMOS, and the techniques necessary to access the Analog-to-Digital Converter. However, the majority of the potential users of the 1800 were non-programmers. Accordingly, it was decided that users should be provided with an Analog-to-Digital Supervisor, hereafter referred to as ADSUP, to facilitate use of the analog capability of the 1800.

Essentially, ADSUP supports the conversion to digital values of any type of voltage waveform data from an experimental apparatus, such as an audio tape recorder. Aside from furnishing the obvious advantages of a system which requires no programming, ADSUP also eliminates redundancy of effort by prospective users of the facility.





## Footnotes

- <sup>1</sup> Wells, Richard A. "The University of Illinois Attached Machine Operating System: Design Criteria and Program Logic," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 334, 1969.



## 2. SYSTEM DESCRIPTION

The function of the 1800 within the System/360 complex is primarily one of data collection. Any subsequent data reduction that a user might wish is better performed on the 360 because floating point and character manipulation instructions are not present on the 1800. Thus, digital values output from the Analog-to-Digital Converter of the 1800 are written onto the 360's nine-track tapes via an 1800 special feature which attaches an 1800 data channel, through a switching unit, directly to a 360 tape control unit. The maximum data rate which the tape unit will accept is 180,000 bytes per second and the 1800 system conversion rate can vary from 9,000 to 24,000 words per second (one 1800 word = two 360 bytes). Therefore, assuming the buffer size is large enough to absorb any operating system overhead, there is no danger of data overrun.

A customized hardware interface, built by the Department, provides for the convenient attachment of user equipment. It consists of an audio tape recorder, 16 analog input terminals, 16 electronic contact operate terminals, 16 process interrupt switches and terminals, and an external sync terminal. Each analog input terminal is an analog multiplexor input point, called a port, and is used for the attachment of an analog input source device. The contact operate terminals are essentially switches which may be used to close circuits on user connected equipment. These switches are activated by the 1800 and are set by program control. Process interrupt switches may be set to



generate an external interrupt to the 1800 for purposes of executing a pre-determined sequence of instructions. The operation of the Analog-to-Digital Converter may be controlled by means of a timing pulse received through the external sync terminal.

For a comprehensive description of the hardware configuration of the 1800, the reader is referred to Department of Computer Science Report No. 280.<sup>1</sup>



## Footnotes

- <sup>1</sup> Wells, R. A., Carter, C. E., Friedman, H. G. "ILLINET Analog Facilities," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 280, 1968.





### 3. DESIGN CRITERIA

Several fundamental requirements, dictated by prospective users' needs and/or desires, guided the design philosophy of ADSUP. Primary among these was providing a system which required no programming effort on the user's part. Therefore, a simple command language was conceived which could be used to specify any possible condition that could be programmed for. ADSUP reads these commands from either the input stream or the console typewriter.

Equally important was supporting the maximum conversion rates of the Analog-to-Digital Converter (ADC). The 1800 provides three control modes for the input of analog data: (1) Programmed Control, (2) Chained Sequential Control, and (3) Random Control. The Programmed mode requires the execution of an I/O command for each digital value read into storage. The overhead imposed by this method makes it extremely slow and therefore unsatisfactory for our purposes.

The Chained Sequential mode requires the execution of only one I/O command to initiate the conversion of many words of data. The data channel continues sampling until stopped by a hardware signal. However, this method allows only sequentially addressed ports (e.g., 7, 8, 9, ...) to be accessed. While this method would support the required conversion rates, the sequential restriction made it necessary to reject it.

The Random mode allows each port to be addressed uniquely in an arbitrarily determined order. It uses two data channels, one to select



multiplexor ports and the other to transmit digitized data from the ADC to core storage. As in the case of the Chained Sequential mode, once an I/O operation has been initiated, the data channel continues it until terminated. Therefore, it was decided to use the Random Control mode of operation.

It was also essential that ADSUP use as little core as possible in order to be able to generously allocate input buffers. Since the logic of ADSUP is naturally separable into two distinct functions, namely, interpreting the user commands and then performing the actual conversion, it was expedient to write the system in two modules: the command processing module and the execution module. Each module loads the other, overlaying itself with its successor, and thus minimizing core utilization. However, since each module may be entered an indefinite number of times, each must be written in such a manner that it is serially re-usable.

Communication between the two modules was accomplished by means of a resident parameter table, called the command options table. It is initially defined with default values for each command. These values are modified by the command phase when other options are chosen. Rather than re-initialize the table each time the command phase is entered, values specified from a previous run remain in effect unless the user specifically overrides them. Thus, only those variables which change from run to run need be given.

Lastly, it was necessary that ADSUP provide a tape file manipulation capability. This was needed in the event a user either



wished to add files to an already existing data tape or rewrite a file containing unwanted data. File labels were also considered to be a useful feature. As a corollary to file manipulation, volume switching was also included to handle cases where a user wished to segregate his output onto separate tapes.



#### 4. PROGRAM LOGIC

As mentioned earlier, ADSUP is composed of a Command phase and an Execute phase. The Command phase was designed to build the command options table in such a manner that the Execute phase would need to do a minimum of initialization before beginning actual conversion. Further, the less core used by the Execute phase, the more core available for I/O buffers. The command options table consists of switches denoting the various options specified by each command, as well as quantitative descriptions, where applicable. As control passes from the Command phase to the Execute phase, and vice versa, core storage is segmented as shown in Figure 1.

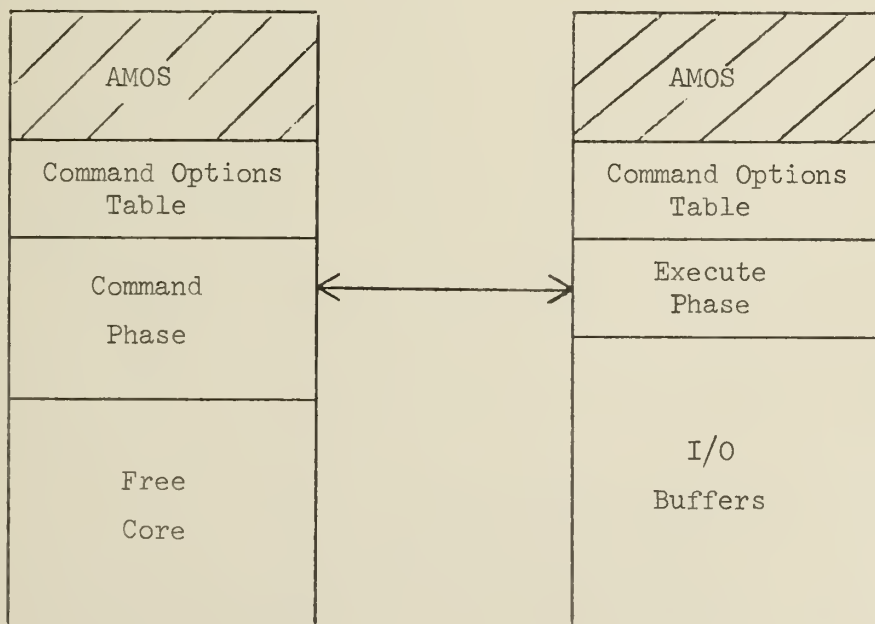


Figure 1.





#### 4.1 Command Phase

The Command phase was written in a modular fashion and primarily consists of vector tables and subroutines. Obviously, this type of program organization lends itself to easy modification. The general flow of the Command phase follows.

After each command is read, it is printed out so the user will have a record of his requested options for each run. Then, the syntax of the command is compared with each entry in a list of legal commands until either a match is found or the list is exhausted. If there is no match, an error message is typed. Otherwise, a branch is taken to the appropriate subroutine for decoding the command. The subroutine fills in the command options table according to the parameters selected and then returns to the driver which obtains the next command. If an error is found in the command, the subroutine prints an explanatory error message. Since the module is table driven, the commands may be input in any order. Also, there is no restriction on the number of times a command may be specified (in the event the user changes his mind on an option specified before entering the Execute phase) since each subroutine is independent of any previously specified command.

A list of the valid commands appears in Figure 2; default options are underlined. The meta-symbols [ ] denote optional parameters. The meta-symbol | indicates that a choice is to be made among options within the meta-symbols { }. Punctuation marks







LABEL, { NONE | CØNSØLE | SYSIN | PRØGRAM }  
 ØVERLØAD, { IGNØRE | LØG | STØP }  
 STATISTICS, { CØNSØLE | SYSØUT | BØTH }  
 ERRØRS, { CØNSØLE | TERMINATE }  
 FILE#, { n | NEXT }  
 REWIND  
 BACKSPACE  
 EXECUTE  
 TERMINATE

Figure 2.--continued

The meaning of each command is presented below.

### INPUT

To define the source of the commands, the INPUT command is provided. This gives the user the ability to direct ADSUP's reader to either the 1816 typewriter (CØNSØLE) or the card input system (SYSIN).

### BLØCKSIZE

The BLØCKSIZE command is used to specify the analog input buffer size. This will also be the size of each tape record written. In order to perform continuous conversion, a double buffering scheme is required. Thus, digitized data is read into one buffer while an already filled buffer is written onto tape. After first verifying that the size requested is not too large for the



available core, ADSUP places the value in the command options table.

### PØRT

The PØRT command specifies one or more multiplexor ports to be sampled. They are sampled in the order designated. In addition to the 16 ports available on the hardware interface, the 1800 has an analog input calibration facility which is selected internally by addressing multiplexor address 13E8<sub>16</sub>. The user may request this by specifying TEST as one of the PØRT parameters. It is treated just like any other port by ADSUP and is sampled at any specified place in the port sequence. This command is required, as there is no default option.

### PRECISIØN

The accuracy required in the digitized data may be specified by the PRECISIØN command. An 1800 data word allows a sign bit plus 14 bits resolution for each digitized value. The 16th bit is used to indicate an overload condition, which will be discussed later. The format of the data words permitted by the 1800 is 8, 11, or 14 significant bits. This is illustrated in Figure 3. The ADC conversion time depends only upon the resolution of the digitized output. Conversion times are as follows: 8 bits, 29 microseconds; 11 bits, 36 microseconds; and 14 bits, 44 microseconds.





	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14 bit format	±	X	X	X	X	X	X	X	X	X	X	X	X	X	X	I
11 bit format	±	X	X	X	X	X	X	X	X	X	X	1	0	0	0	I
8 bit format	±	X	X	X	X	X	X	X	X	1	0	0	0	0	0	I

Notes:

1. Bit 0 contains the sign bit of the data.
2. The X's may be zero or one, depending on the converted value.
3. The 1 in bit position 9 of the 8-bit format and in bit position 12 of the 11-bit format is provided for half-adjust of the quantizing error.
4. I is the overload indicator. It is 1 if an overload condition exists.

Figure 3.

DELAY

The DELAY option may be used when a user does not wish to convert at full internal speed, but rather wants samples taken only at particular times. If EXTERNAL is specified, conversion proceeds under control of external sync, and thus occurs only when a start pulse is provided. One port is sampled for each sync pulse generated. If the TIMER parameter is specified, the sequence of



ports defined in the ~~P~~ORT command is sampled once every  $n$  milliseconds, where  $n \leq 4096$  milliseconds. With the ~~P~~RCESS option, the port sequence is sampled once each time a process interrupt is received from bit  $m$ . When ~~N~~ONE is specified, conversion proceeds at the maximum rate of the ADC.

### RECORDER

The ~~REC~~ORDER command is used to control the starting of the analog source device, usually a tape recorder. If ~~O~~PERATE is specified, ADSUP will activate contact operate bit  $n$ , thus closing the circuit and turning on the apparatus attached to the terminal. The user may delay the beginning of conversion  $m$  milliseconds ( $m \leq 4096$  milliseconds) if he wishes some warmup time to allow his tape recorder to reach maximum operating speed.

### START

To indicate to ADSUP how the user wishes to initiate sampling in the Execute phase, the START command is provided. If ~~AUT~~OMATIC is specified, conversion begins when the Execute phase is entered. If the ~~P~~RCESS parameter is chosen, conversion begins when a process interrupt is received from bit  $n$ . The typewriter ~~C~~ONSOLE may also be used to start sampling; conversion begins when the message ~~G~~O is typed.



STOP

Conversion may be terminated by one of four means, and this choice is communicated to ADSUP by means of the STOP command. By specifying the TIME or BLOCKS parameter, conversion may be terminated after m seconds or after having converted n blocks of data, respectively. For more dynamic control, it may end when a process interrupt is received from bit k if the PROCESS option is requested. Or, if CONSOLE is specified, conversion ends when the user types STOP.

VOLID

The VOLID command is used to specify the external reel number of the desired output tape(s). This option may be used to direct output to different tapes for each run. This command is required.

OVERLOAD

The allowable voltage range of the ADC is  $\pm 5$  volts. The maximum positive voltage has a digitized value of 16,832 (using 14-bit resolution). The maximum negative value is 16,383. If a voltage input is outside the  $\pm 5$  volt range, an overload condition exists. The user specifies the handling of this condition by means of the OVERLOAD command. If he wishes execution terminated, he indicates STOP. He may also IGNORE all overloads, or have each



occurrence counted and communicated to him at the end of the run by means of the LOG option.

### STATISTICS

By means of the STATISTICS command, the user can direct the listing of the run statistics to either the 1816 typewriter (CONSOLE), the system output stream (SYSOUT), or BOTH. The statistics provided by ADSUP are: file number written, the time conversion began and ended, the blocksize used, number of blocks written, and, optionally, the number of overloads encountered.

### ERRORS

The ERRORS command is used to direct ADSUP's action upon finding an error within the Command phase. The user may elect to TERMINATE processing, or an error message may be typed on the CONSOLE and the corrected command may be re-entered.

### REWIND, BACKSPACE, FILE#

The tape handling commands provided are REWIND, BACKSPACE, and FILE#. REWIND is used to reposition the tape at load point. To position the tape at the beginning of the last record written, the BACKSPACE command is used. The FILE# command provides the ability to position the tape at any file desired. Using only the NEXT parameter, the tape files are written sequentially.





LABEL

The LABEL command is used to specify the source of user labels. Each tape file written is preceded by an 80 byte ADSUP label, which contains the word AMOS, the file number, and the date and time of creation. Optionally, there may be another 80 byte label following the required one. This label may be copied from a card in the input stream (SYSIN) or may be typed on the 1816 typewriter (CØNSØLE). Alternatively, the label may be created dynamically by the calling PRØGRAM. In this case, ADSUP exits to the user's program, passing it the address of an area in which the user places the label. If no label is desired, NØNE is specified.

EXECUTE

To leave the Command phase and enter the Execute phase is the purpose of the Execute command. A software switch is interrogated to determine whether the Execute phase should be loaded. If any errors were detected by the other command subroutines and the ERRØRS option specified was TERMINATE, the switch is on and ADSUP therefore exits to the calling program. If the switch is off, the Command options table is checked for errors that could not have been detected earlier.

First, the EXECUTE subroutine examines the tape manipulation commands to make sure that mutually exclusive commands (e.g., REWIND and BACKSPACE) have not been specified. Then, process interrupt bit specifications are checked for duplication of function. For example,



the same process interrupt switch cannot be used both to stop conversion and to dictate sampling delays. Following this, the port entry in the options table is checked to verify that at least one port was specified. Lastly, the status of the output tape is checked and, if this is the initial reference to that tape, AMOS is passed pertinent control information. The EXECUTE command is required.

#### TERMINATE

The function of the TERMINATE command is merely to conclude all processing by ADSUP and to return to the calling program. If input is the card stream, an end-of-file is treated as a TERMINATE command.



## 4.2 Execute Phase

This section describes the generalized logic flow of the Execute phase. Initially, the discussion will deal with the case in which the DELAY option requested was either ~~NONE~~ or EXTERNAL. The effect of the ~~PROCESS~~ and TIMER options will be discussed later.

I/O on the 1800 is controlled by a two word I/O Control Command (IOCC). The IOCC defines the I/O operation, I/O device, core storage address for data, and associated modifiers relative to the I/O operation (e.g., external sync, data word resolution, control mode for analog I/O). The modifier field is dynamically created by ADSUP based on the values in the command options table entries for ~~PRECISION~~ and DELAY. The analog control mode is set for Random Control.

This mode utilizes two 1800 data channels, one to read digitized data from the ADC into core and one to send to the converter a list of the multiplexor port addresses to be sampled. The Multiplexor Port Address Table (MPAT) is created dynamically by repetitively storing the sequence of ports specified by the ~~PORT~~ command into a buffer whose length is equal to the ~~BLOCKSIZE~~. That is, if the user's requests were ~~PORT~~ 1,2,3 and ~~BLOCKSIZE~~,450, then the MPAT would contain 450 entries in the following order: 1,2,3,1,2,3,1,...,1,2,3. In order for conversion to proceed indefinitely, port addresses must be supplied to the ADC indefinitely also. To accomplish this, the MPAT is chained to itself.

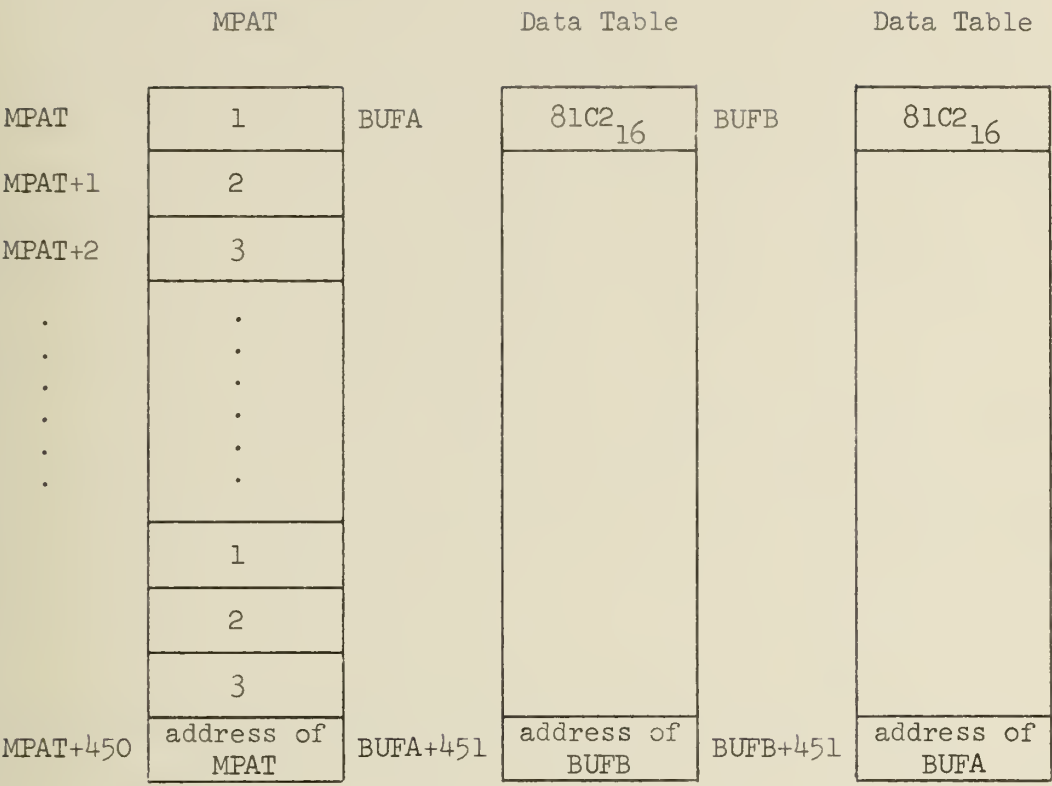


Furthermore, to keep up with the conversion rates, the two I/O buffers must be cross-chained so that upon filling one buffer, the hardware knows to continue reading data into the other. The buffers are known as data tables. Each contains a control word, called the word count, needed to pass information to the hardware. The word count defines the number of words to be read into the data table and the action to be taken upon filling a data table. The latter function is handled by two scan control bits in the word count. One bit describes the type of scan desired: single scan of the data table or continuous scan of more than one. The other bit is used to tell the hardware whether or not to generate an interrupt at the end of a data table.

The number of words in a data table is obtained from the options table ~~BL~~CKSIZE entry. The scan control bits are set for continuous scan with end-of-table interrupt. For the example given in the preceding paragraph, the MPAT and data tables would be represented as in Figure 4.







Note: The first word in the data table is the word count. The high order two bits are the scan control bits, and are set for continuous scan and end-of-table interrupt.

Figure 4.

Based on process interrupt bit assignments, a process interrupt branch table is created whose entries consist of the addresses of routines for handling START, STOP, and DELAY interrupts. Entries for unassigned bits contain the address of an error routine. Upon being notified by AMOS of a process interrupt, ADSUP branches



to the appropriate routine by indexing from the table origin a displacement equal to the process interrupt bit number.

After creating the table, any necessary tape positioning is done and an internal file counter is modified accordingly. Then, the AMOS label is written at the beginning of the file, followed optionally, by the user supplied label.

Once this processing has been completed, ADSUP is ready to begin converting. This starts immediately unless the user has specified otherwise in the Command phase, in which case ADSUP waits for the user's start signal.

Conversion proceeds by issuing two I/O commands. One (Initialize Write) sends the MPAT to the ADC, and the other (Initialize Read) sends converted data into the first buffer. ADSUP then waits for an I/O interrupt from the ADC. If an interrupt occurs due to an overload, it is handled according to the ~~OVERLOAD~~ specification in the command options table. Then, ADSUP waits for another I/O interrupt. When an end-of-table interrupt occurs, the buffer is written out on tape. If the interrupt was neither of the above, then it is a serious hardware malfunction, and an error message is typed out and conversion is terminated.

It should be recalled that the ADC does not stop converting when a data table is filled. After signaling the 1800 program of an end-of-table condition, it continues reading digitized data into the other data table. While this data table is being filled, the contents of the first one are being written onto tape. Therefore, to avoid



having data overwritten, it is imperative that the first data table be written to tape before the second one is filled from the ADC. To police the problem, ADSUP checks the status of the previous tape operation before issuing a new write request. If the last operation has not completed, a buffer overflow conditions exists. An error message is typed out and conversion is terminated. The user may remedy this situation by increasing his blocksize request.

After initializing the output to tape, ADSUP checks to see if the user-requested stop condition has been met. If it has occurred, conversion is terminated. If not, the next ADC I/O interrupt is awaited.

The description to this point has covered the program logic for the case where DELAY is ~~NONE~~ or EXTERNAL. TIMER or ~~PROCESS~~ delays must be handled somewhat differently, since conversion is not proceeding at full speed.

In the first place, the length of the MPAT is equal to the number of ports specified and the MPAT is not chained to itself, since the port sequence is sampled only once each time the start signal (timer or process interrupt) occurs. Also, the data table length is set to the same length as the MPAT. The scan control bits are set for a single scan, end-of-table interrupt. After receiving the interrupt, the digitized data is moved into the large buffer for tape output. A double buffering scheme is still required since even with the delay, conversion speed can approach internal speed closely enough that the danger of buffer overflow still exists. The only



other deviation from the logic described before is that the stop signal would probably occur while an output buffer was only partially filled. To allow immediate response to the stop request, it was decided to write out the partially filled buffer, rather than waiting for it to be filled.

Regardless of the DELAY option specified, once the ADC has been stopped, statistics from the run are listed according to the STATISTICS command. Then, constants are re-initialized to insure that the module is serially re-usable. Lastly, control is returned to the Command phase in the event the user wishes to convert further data. Final exit from ADSUP is made only from the Command phase upon encountering the TERMINATE command. Thus the user may re-enter each phase an indefinite number of times.

In conclusion, experience has shown that it is dangerous to assume that a system program of any complexity is ever finished. Time, user sophistication, and new user needs demand that changes be made. Consequently, during development of ADSUP the utmost consideration was given to insuring flexibility. Within the existing framework, it is easy to add or delete commands and to modify existing ones.





## BIBLIOGRAPHY

- "ADSUP User's Guide," Department of Computer Science, University of Illinois, Urbana, Illinois, 1969.
- "Attached Machine Operating System User's Guide, Version 2.2," Department of Computer Science, University of Illinois, Urbana, Illinois, 1968.
- "IBM 1800 Assembler Language," IBM Systems Reference Library, Form No. C26-5882, International Business Machines Corporation, San Jose, California, 1968.
- "IBM 1800 Functional Characteristics," IBM Systems Reference Library, Form No. A26-5918, International Business Machines Corporation, San Jose, California, 1968.
- Wells, R. A., Carter, C. E., and Friedman, H. G. "ILLINET Analog Facilities," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 280, 1968.
- Wells, Richard A. "The University of Illinois Attached Machine Operating System: Design Criteria and Program Logic," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 334, 1969.



NOV 28 1972















UNIVERSITY OF ILLINOIS-URBANA



3 0112 056690867